UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/607,591	06/27/2003	Mark Ronald Plesko	3382-64706	5996

26119 7590 02/09/2009

KLARQUIST SPARKMAN LLP
121 S.W. SALMON STREET
SUITE 1600
PORTLAND, OR 97204

EXAMINER	
DAO, THUY CHAN	
ART UNIT	PAPER NUMBER
2192	
MAIL DATE	DELIVERY MODE
02/09/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/607,591	PLESKO ET AL.	
	Examiner	Art Unit	
	Thuy Dao	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 December 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-7,9,10,12-24 and 26-32* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-7,9,10,12-24 and 26-32* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *27 June 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *09/04/08 and 12/08/08*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the amendment filed on December 8, 2008.

2. Claims 1, 3-7, 9, 10, 12-24 and 26-32 have been examined.

Response to Amendments

3. In the instant amendment, claims 14 and 28-29 have been amended.

4. The 35 USC §101 rejection over claims 14, 15, 17-23, 28, 29 and 32 is withdrawn in view of Applicant's amendments.

Response to Arguments

5. Applicants' arguments have been considered.

a) The limitations at issue "*...the plurality of different source languages comprise ... at least one untyped source language*" (Remarks, page 10).

As clearly set forth in the prior Office action mailed September 8, 2008, page 5, Gordon teaches *the plurality of different source languages comprise ... at least one untyped source language* (e.g., col.35: 8-20, an untyped source language such as Smalltalk).

As well-known in the art, Smalltalk is an untyped source language. The examiner respectfully directs Applicants' attention to:

"Smalltalk: A White Paper Overview" to Porter III, published March 2003, which explicitly teaches Smalltalk as an untyped source language (e.g., pages 15 and 40); and

US Patent No. 6,041,179, which also teaches Smalltalk as an untyped source language (e.g., col.3: 35-49).

These prior art have been made of record and attached herein in this Office action.

b) The limitations at issue "*... a type designated as an unknown type, wherein the unknown type indicates that an element of the representation is of a type that is not known*" (e.g., claim 1, lines 9-10).

Applicant's arguments have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground of rejection is made in view of Leach et al.- US Patent No. 6,412,020 as set forth in detail below.

Claim Rejections – 35 USC §103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1, 3-7, 9-10, 12-24, and 26-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gordon (art of record, US Patent No. 6,560,774) in view of Lidin (art of record, "Inside Microsoft .NET IL Assembler") and US Patent No. 6,412,020 to Leach et al. (art made of record, hereafter "Leach").

Claim 1:

Gordon discloses *a method of type-checking a code segment written in a programming language comprising:*

translating the code segment from the programming language to one or more representations of an intermediate language (e.g., FIG. 2, col.6: 8-57),

wherein the one or more representations of the intermediate language are capable of representing programs written in a plurality of different source languages (e.g., FIG. 23, col.27: 4-33),

wherein the plurality of different source languages comprise at least one typed source language (e.g., FIG. 2, Visual Basic VB, Visual C++ VC++, col.6: 8-57) *and*

at least one untyped source language (e.g., SmallTalk, col.35: 8-20)*; and*

type-checking the one or more representations based on a rule set (e.g., col.1: 54-62; col.6: 58 – col.7: 16; col.26: 5-14),

wherein the rule set comprises rules for type-checking a type designated as an known type (e.g., FIG. 24, col.27: 35 – col.28: 29; col.23: 46 – col.24: 49),

wherein the known type indicates that an element of the representation is of a type that is known (e.g., col.17: 52 – col.18: 28; col.20: 44-57; col.27: 35 – col.28: 29).

In an analogous art, Lidin further discloses rules for type-checking (e.g., chapter 7, Table 7-6, Native Types Defined in the Runtime) and a native type named "IUNKNOWN" (e.g., chapter 7, page 10).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Lidin's teaching into Gordon's teaching. One would have been motivated to do so to denote native types and perform type-checking in .NET framework in the runtime as suggested by Lidin (e.g., pp. 9-10 and 12-13).

Neither Gordon nor Lidin explicitly discloses IUNKNOWN as a type *designated as an unknown type, wherein the unknown type indicates that an element of the representation is of a type that is not known.*

However, in an analogous art, Leach further discloses IUNKNOWN as *a type designated as an unknown type, wherein the unknown type indicates that an element of the representation is of a type that is not known* (e.g., col.6: 59 – col.7: 18, "IUNKNOWN" is the unknown type indicating that an element (interface) of IUNKNOWN is unknown).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Leach's teaching into Gordon and Lidin's teaching. One would have been motivated to do so to perform type-checking/determine whether an object to be embedded supports a specific interface as suggested by Leach (e.g., col.6: 59-67).

Claim 3:

The rejection of claim 1 is incorporated. Gordon discloses *the rule set is selected from a plurality of rule sets* (e.g., col.25: 51 – col.26: 25).

Claim 4:

The rejection of claim 3 is incorporated. Leach discloses *only a fraction of the plurality of rule sets contain rules for type-checking a type designated as an unknown type, wherein the unknown type indicates that an element of the representation is of a type that is not known* (e.g., col.6: 59 – col.7: 18).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Leach's teaching into Gordon and Lidin's teaching. One would have been motivated to do so to perform type-checking/determine whether an object to be embedded supports a specific interface as suggested by Leach (e.g., col.6: 59-67).

Claim 5:

The rejection of claim 1 is incorporated. Gordon discloses *the rule set further comprises rules for type-checking types representing categories of types found in a plurality of programming languages* (e.g., FIG. 23, col.27: 4-33).

Claim 6:

Gordon discloses *a method of selectively retaining type information during compilation in a code segment written in a programming language, the method comprising:*

translating the code segment from the programming language to one or more representations of an intermediate language (e.g., FIG. 2, col.6: 8-57)*;*

for each representation, determining whether to retain type information for one or more elements of the representation (e.g., FIG. 23, col.27: 4-33)*;*

based on the determination, associating one or more elements of the representation with a type (e.g., col.1: 54-62; col.6: 58 – col.7: 16; col.26: 5-14),

designated as an known type, indicating the element can be of any type; and type-checking the one or more representations based on a rule set (e.g., FIG. 24, col.27: 35 – col.28: 29; col.23: 46 – col.24: 49),

wherein the rule set comprises rules for type-checking the type designated as the known type (e.g., col.17: 52 – col.18: 28; col.20: 44-57; col.27: 35 – col.28: 29).

In an analogous art, Lidin further discloses rules for type-checking (e.g., chapter 7, Table 7-6, Native Types Defined in the Runtime) and a native type named "IUNKNOWN" (e.g., chapter 7, page 10).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Lidin's teaching into Gordon's teaching. One would have been motivated to do so to denote native types and perform type-checking in .NET framework in the runtime as suggested by Lidin (e.g., pp. 9-10 and 12-13).

Neither Gordon nor Lidin explicitly discloses IUNKNOWN as a type *designated as an unknown type, wherein the unknown type indicates that an element of the representation is of a type that is not known.*

However, in an analogous art, Leach further discloses IUNKNOWN as *a type designated as an unknown type, wherein the unknown type indicates that an element of the representation is of a type that is not known* (e.g., col.6: 59 – col.7: 18, "IUNKNOWN" is the unknown type indicating that an element (interface) of IUNKNOWN is unknown).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Leach's teaching into Gordon and Lidin's teaching. One would have been motivated to do so to perform type-checking/determine whether an object to be embedded supports a specific interface as suggested by Leach (e.g., col.6: 59-67).

Claim 7:

The rejection of claim 6 is incorporated. Gordon discloses *the determination is based on a current stage of compilation, a characteristic of each representation, or the programming language* (e.g., col.6: 45 – col.7: 34; col.15: 37-67).

Claim 9:

The rejection of claim 6 is incorporated. Leach discloses *the type, designated as the unknown type, indicating the element can be of any type has size information associated with it* (e.g., e.g., col.6: 59 – col.7: 18).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Leach's teaching into Gordon and Lidin's teaching. One would have been motivated to do so to perform type-checking/determine whether an object to be embedded supports a specific interface as suggested by Leach (e.g., col.6: 59-67).

Claim 10:

The rejection of claim 9 is incorporated. Leach discloses *generating code from at least elements associated with the type, designated as the unknown type, indicating the element can be of any type based on the size information* (e.g., col.6: 59 – col.7: 18).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Leach's teaching into Gordon and Lidin's teaching. One would have been motivated to do so as set forth above.

Claim 12:

Gordon discloses *a method of translating types associated with a plurality of programming languages to types of an intermediate language, the method comprising:*

replacing the types associated with the plurality of programming languages with the types of the intermediate language (e.g., FIG. 2, col.6: 8-57),

wherein the types of the intermediate language comprise general categories of the types (e.g., col.17: 52 – col.18: 28; col.20: 44-57; col.27: 35 – col.28: 29)

associated with the plurality of programming languages and a type designated as an unknown type (e.g., FIG. 23, col.27: 4-33),

wherein the type designated as the unknown type has size information associated with it (e.g., col.1: 54-62; col.6: 58 – col.7: 16; col.26: 5-14),

wherein the size information comprises size information of a machine representation of the type designated as the unknown type (e.g., FIG. 24, col.27: 35 – col.28: 29; col.23: 46 – col.24: 49).

In an analogous art, Lidin further discloses rules for type-checking (e.g., chapter 7, Table 7-6, Native Types Defined in the Runtime) and a native type named "IUNKNOWN" (e.g., chapter 7, page 10).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Lidin's teaching into Gordon's teaching. One would have been motivated to do so to denote native types and perform type-checking in .NET framework in the runtime as suggested by Lidin (e.g., pp. 9-10 and 12-13).

Neither Gordon nor Lidin explicitly discloses IUNKNOWN as a type *designated as an unknown type, wherein the unknown type indicates that an element of the representation is of a type that is not known.*

However, in an analogous art, Leach further discloses IUNKNOWN as *a type designated as an unknown type, wherein the unknown type indicates that an element of the representation is of a type that is not known* (e.g., col.6: 59 – col.7: 18, "IUNKNOWN" is the unknown type indicating that an element (interface) of IUNKNOWN is unknown).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Leach's teaching into Gordon and Lidin's teaching. One would have been motivated to do so to perform type-checking/determine whether an object to be embedded supports a specific interface as suggested by Leach (e.g., col.6: 59-67).

Claim 13:

The rejection of claim 12 is incorporated. Gordon discloses *the types of the intermediate language further comprise types related to programming language specific primitive types* (e.g., col.16: 59 – col.17: 42).

Claim 14:

Gordon discloses *a system for type-checking an intermediate representation of source code in a compiler comprising:*

one or more types associated with elements of the intermediate representation (e.g., FIG. 2, col.6: 8-57),

wherein at least one of the types, designated as an unknown type, indicates an element can be of any type (e.g., FIG. 23, col.27: 4-33; col.17: 52 – col.18: 28; col.20: 44-57; col.27: 35 – col.28: 29);

one or more rule sets comprising rules associated with the type (e.g., col.1: 54-62; col.6: 58 – col.7: 16; col.26: 5-14),

designated as the unknown type, indicating an element can be of any type (e.g., FIG. 24, col.27: 35 – col.28: 29; col.23: 46 – col.24: 49); *and*

a type-checker for applying the one or more rule sets to the elements of the intermediate representation (e.g., col.17: 52 – col.18: 28; col.20: 44-57; col.27: 35 – col.28: 29).

In an analogous art, Lidin further discloses rules for type-checking (e.g., chapter 7, Table 7-6, Native Types Defined in the Runtime) and a native type named "IUNKNOWN" (e.g., chapter 7, page 10).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Lidin's teaching into Gordon's teaching. One would have been motivated to do so to denote native types and perform type-checking in .NET framework in the runtime as suggested by Lidin (e.g., pp. 9-10 and 12-13).

Neither Gordon nor Lidin explicitly discloses IUNKNOWN as a type *designated as an unknown type, wherein the unknown type indicates that an element of the representation is of a type that is not known.*

However, in an analogous art, Leach further discloses IUNKNOWN as *a type designated as an unknown type, wherein the unknown type indicates that an element of the representation is of a type that is not known* (e.g., col.6: 59 – col.7: 18,

"IUNKNOWN" is the unknown type indicating that an element (interface) of IUNKNOWN is unknown).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Leach's teaching into Gordon and Lidin's teaching. One would have been motivated to do so to perform type-checking/determine whether an object to be embedded supports a specific interface as suggested by Leach (e.g., col.6: 59-67).

Claim 15:

The rejection of claim 14 is incorporated. Leach discloses *the type, designated as the unknown type, indicating the element can be of any type has size information associated with it* (e.g., col.6: 59 – col.7: 18).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Leach's teaching into Gordon and Lidin's teaching. One would have been motivated to do so as set forth above.

Claim 17:

The rejection of claim 14 is incorporated. Gordon discloses *the one or more rule sets applied to the elements of the intermediate representation are selected based on the stage of compilation* (e.g., col.19: 60 – col.20: 10; col.23: 46 – col.24: 10).

Claim 18:

The rejection of claim 14 is incorporated. Gordon discloses *the one or more rule sets applied to the elements of the intermediate representation are selected based on a characteristic of the source code* (e.g., col.34: 41-64; col.35: 8-44).

Claim 19:

The rejection of claim 14 is incorporated. Gordon discloses *the one or more rule sets applied to the elements of the intermediate representation are selected based on the intermediate representation* (e.g., col.25: 51 – col.26: 25; col.35: 50 – col.36: 12).

Claim 20:

The rejection of claim 14 is incorporated. Leach discloses *only a fraction of the one or more rule sets contain rules for type-checking a type, designated as an unknown type, that indicates an element can be of any type* (e.g., e.g., col.6: 59 – col.7: 18).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Leach's teaching into Gordon and Lidin's teaching. One would have been motivated to do so as set forth above.

Claim 21:

The rejection of claim 14 is incorporated. Gordon discloses *the one or more rule sets further comprise rules for type-checking types representing categories of types found in a plurality of programming languages* (e.g., col.27: 4-33).

Claim 22:

The rejection of claim 14 is incorporated. Gordon discloses *the system selectively retains type information for some elements of the intermediate representation and selectively does not retain type information for other elements of the intermediate representation* (e.g., col.17: 53 – col.18: 28; col.19: 60 – col.20: 10).

Claim 23:

The rejection of claim 22 is incorporated. Leach discloses *the system selectively does not retain type information for an element of the intermediate representation by replacing a type associated with the element with the type, designated as the unknown type, indicating an element can be of any type* (e.g., e.g., col.6: 59 – col.7: 18).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Leach's teaching into Gordon and Lidin's teaching. One would have been motivated to do so as set forth above.

Claim 24:

Gordon discloses *a method of representing types in an intermediate language comprising:*

defining a plurality of types to be associated with elements of the intermediate language (e.g., FIG. 2, col.6: 8-57),

wherein one of the plurality of types indicates that an element of the intermediate language is associated with a type designated as an known type (e.g., FIG. 23, col.27: 4-33; col.17: 52 – col.18: 28; col.20: 44-57; col.27: 35 – col.28: 29);

wherein the type indicating that an element of the intermediate language is associated with the type designated as the known type has a size associated with it (e.g., col.1: 54-62; col.6: 58 – col.7: 16; col.26: 5-14),

wherein the size represents size of a machine representation of the type designated as the known type (e.g., FIG. 24, col.27: 35 – col.28: 29; col.23: 46 – col.24: 49).

In an analogous art, Lidin further discloses rules for type-checking (e.g., chapter 7, Table 7-6, Native Types Defined in the Runtime) and a native type named "IUNKNOWN" (e.g., chapter 7, page 10).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Lidin's teaching into Gordon's teaching. One would have been motivated to do so to denote native types and perform type-checking in .NET framework in the runtime as suggested by Lidin (e.g., pp. 9-10 and 12-13).

Neither Gordon nor Lidin explicitly discloses IUNKNOWN as a type *designated as an unknown type, wherein the unknown type indicates that an element of the representation is of a type that is not known.*

However, in an analogous art, Leach further discloses IUNKNOWN as *a type designated as an unknown type, wherein the unknown type indicates that an element of the representation is of a type that is not known* (e.g., col.6: 59 – col.7: 18, "IUNKNOWN" is the unknown type indicating that an element (interface) of IUNKNOWN is unknown).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Leach's teaching into Gordon and Lidin's teaching.

One would have been motivated to do so to perform type-checking/determine whether an object to be embedded supports a specific interface as suggested by Leach (e.g., col.6: 59-67).

Claim 26:

The rejection of claim 24 is incorporated. Leach discloses *an element of the intermediate language that was previously associated with another type is associated with the type indicating that the element is associated with the type designated as the unknown type* (e.g., col.6: 59 – col.7: 18).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Leach's teaching into Gordon and Lidin's teaching. One would have been motivated to do so as set forth above.

Claim 27:

The rejection of claim 24 is incorporated. Gordon discloses *the plurality of types further comprises types representing categories of types found in a plurality of programming languages* (e.g., col.6: 8-57; col.15: 37-67).

Claim 28:

Claim 28 is a computer-readable storage medium version, which recite(s) the same limitations as those of claim 24, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the reference teaches all of the limitations of the above claim(s), it also teaches all of the limitations of claim 28.

Claim 29:

Claim 29 is a computer-readable storage medium version, which recite(s) the same limitations as those of claim 1, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the reference teaches all of the limitations of the above claim(s), it also teaches all of the limitations of claim 29.

Claim 30:

The rejection of claim 1 is incorporated. Leach discloses *the rule set further comprises rules for dropping type information for one or more elements of the representation by changing a known type of the one or more elements to the type designated as the unknown type* (e.g., col.6: 59 – col.7: 18).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Leach's teaching into Gordon and Lidin's teaching. One would have been motivated to do so as set forth above.

Claim 31:

The rejection of claim 6 is incorporated. Leach discloses *at least one of the one or more representations of the intermediate language supports dropping type information by designating a type as an unknown type* (e.g., col.6: 59 – col.7: 18).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Leach's teaching into Gordon and Lidin's teaching. One would have been motivated to do so as set forth above.

Claim 32:

The rejection of claim 15 is incorporated. Leach discloses *the size information comprises size information of a machine representation of the type designated as the unknown type* (e.g., col.6: 59 – col.7: 18).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Leach's teaching into Gordon and Lidin's teaching. One would have been motivated to do so as set forth above.

Conclusion

8. Any inquiry concerning this communication should be directed to examiner Thuy Dao (Twee), whose telephone/fax numbers are (571) 272 8570 and (571) 273 8570, respectively. The examiner can normally be reached on every Tuesday, Thursday, and Friday from 6:00AM to 6:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam, can be reached at (571) 272 3695.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273 8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is (571) 272 2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Thuy Dao/
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192